**Amendments to the Claims**

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of claims:

1. (currently amended) A method of masking a conditional jump operation in a cryptographic processor, programmed to execute a sequence of instructions, wherein the conditional jump is determined by evaluating a distinguishing value V against a reference value and wherein the reference value is bounded by an upper limit Vmax and a lower limit Vmin, the method comprising the steps of:
(a) determining a location of a conditional jump in a program; and
(b) inserting processor instruction at said location to direct program execution to one of two branches, said processor instructions computing a target address, the target address being derived from said distinguishing value and a base address constituted by a random number, wherein for each evaluation of said distinguishing value against said reference value a different number of instructions are executed for each conditional jump.

2. (original) A method as defined in claim 1, said distinguishing value being combined with a random value, thereby adding a random number of instructions on every conditional evaluation.

3. (original) A method as defined in claim 1, said inserted instructions including calls to respective subroutines, said subroutines including instructions for changing the return address of the subroutines to said one of two branches.

4. (original) A method as defined in claim 1, said target address is comprised of said distinguishing value V and a random number.

5. (original) A method as defined in claim 4, said target address is computed using an extended addressing mode of said processor.

6. (original) A method of masking a conditional jump operation in a cryptographic processor programmed to execute a sequence of instructions, wherein the conditional jump chooses one of a plurality of execution branches for execution by comparing a distinguishing value V to a reference value, the method comprising the steps of
(a) associating each of the branches with a respective set of addresses;
(b) computing a target address derived from the value V and said reference value, said target address being located in one of said sets of addresses; and
(c) following the instructions at the target address, said instruction directing program execution to the branch associated with said one of said sets of addresses.

7. (original) A method according to claim 6, wherein each set of addresses contains a plurality of addresses.

8. (original) A method according to claim 6, said instructions at each said target address within a set comprising identical instructions each directing execution to said branch associated with the set.

9. (original) A method according to claim 6, said instructions in a set being sequential operations and random address.

10. (original) A method according to claim 6, wherein said instructions at said target address are followed by means of an extended addressing mode of said processor.

11. (canceled)

12. (canceled)

13. (canceled)

14. (canceled)

15. (canceled)

16. (canceled)

17. (canceled)

18. (canceled)

19. (canceled)

20. (canceled)

21. (canceled)

22. (canceled)

23. (canceled)

24. (canceled)

25. (canceled)

26. (new) A method of masking a conditional jump operation (a, f) in a cryptographic processor programmed to execute a sequence of instructions, wherein the conditional jump chooses one of a plurality of execution branches (d, e) for execution based upon a distinguishing value (V), either by comparing with a reference value (TH, THRESHOLD), or by restricting the possible values of said distinguishing value (Vmin, Vmax), the method characterized by the steps of:

(a) associating each of the branches (d, e, i, j) with a respective set of addresses (56, 58);
(b) computing a target address (b, g) from the distinguishing value (V) and either said reference value (THRESHOLD) or a base address (KNOWN_DISPLACEMENT), said target address being located in one of said sets of addresses (56, 58); and
(c) following the instructions at the target address, said instructions directing program execution to the branch (d, e, i, j) associated with said one of said sets of addresses (56, 58).